

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-242505

(43)Date of publication of application : 08.09.2000

(51)Int.Cl.

G06F 9/45

G06F 9/46

(21)Application number : 11-047330

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 25.02.1999

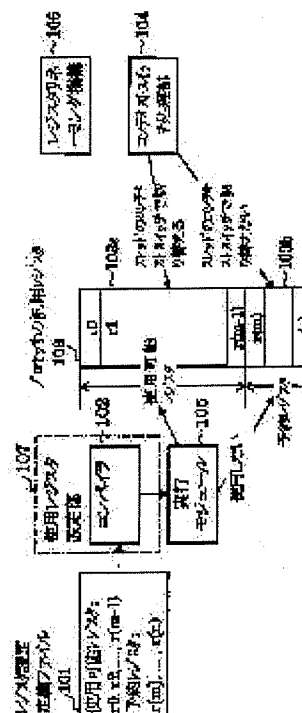
(72)Inventor : OCHIAI SHINICHI

(54) PROCESSING SYSTEM FOR OPERATING SYSTEM AND PROCESSING METHOD FOR OPERATING SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To reduce the overhead time of context switch processing.

SOLUTION: In this processing system for an operating system having plural general purpose registers in which a thread executing processing is available, a register setting definition file 101 which defines a register to be used among the general purpose registers, a use register setting part 107 which reads the file 101 and performs setting so that the thread can use the register defined by the file 101 and a context switch processing part 104 which performs context switch processing that saves and returns data stored in the register which is defined by the file 101 and is used are provided. Thus, the number of registers performing saving and returning is limited, and the context switch processing is made efficient.



LEGAL STATUS

[Date of request for examination]

14.02.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開2000-242505

(P2000-242505A)

(43)公開日 平成12年9月8日(2000.9.8)

(51)Int.Cl. ⁷	識別記号	F I	テマコード [*] (参考)
G 0 6 F 9/45		G 0 6 F 9/44	3 2 2 H 5 B 0 8 1
9/46	3 1 3	9/46	3 1 3 A 5 B 0 9 8

審査請求 未請求 請求項の数11 OL (全 25 頁)

(21)出願番号 特願平11-47330

(22)出願日 平成11年2月25日(1999.2.25)

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 落合 真一

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(74)代理人 100102439

弁理士 宮田 金雄 (外2名)

Fターム(参考) 5B081 CC25

5B098 DD01 GA02 GA05 GA07 GD03

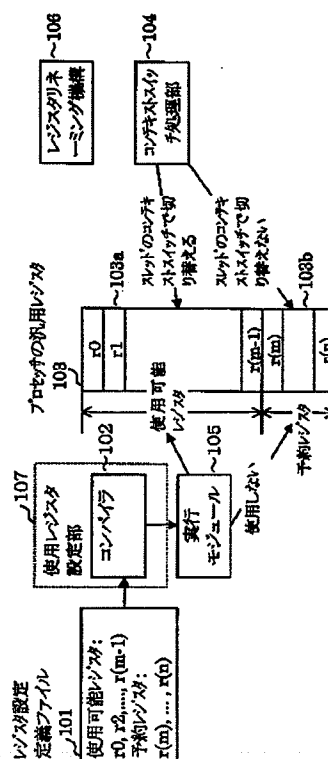
GD12 GD14 GD21

(54)【発明の名称】 オペレーティングシステムの処理方式及びオペレーティングシステムの処理方法

(57)【要約】

【課題】 コンテキストスイッチ処理のオーバーヘッド時間の短縮を図る。

【解決手段】 処理を実行するスレッドが使用可能な複数の汎用レジスタを有するオペレーティングシステムの処理方式において、汎用レジスタのうち使用するレジスタを定義するレジスタ設定定義ファイルと、上記レジスタ設定定義ファイルを読み込み、スレッドが上記レジスタ設定定義ファイルで定義されたレジスタを使用するように設定する使用レジスタ設定部と、上記レジスタ設定定義ファイルで定義された使用するレジスタに記憶されたデータの待避、復帰を行なうコンテキストスイッチ処理部とを備えたことにより、待避、復帰を行なうレジスタの数を制限することによりコンテキストスイッチ処理の効率化を図る。



【特許請求の範囲】

【請求項1】 処理を実行するスレッドが使用可能な複数の汎用レジスタを有するオペレーティングシステムの処理方式において、

汎用レジスタのうち使用するレジスタを定義するレジスタ設定定義ファイルと、

上記レジスタ設定定義ファイルを読み込み、スレッドが上記レジスタ設定定義ファイルで定義されたレジスタを使用するように設定する使用レジスタ設定部と、

上記レジスタ設定定義ファイルで定義された使用するレジスタに記憶されたデータの待避、復帰を行なうコンテキストスイッチ処理を行なうコンテキストスイッチ処理部とを備えたことを特徴とするオペレーティングシステムの処理方式。

【請求項2】 上記レジスタ設定定義ファイルは、定義された使用可能なレジスタを使用可能レジスタとして定義し、上記使用可能レジスタ以外のレジスタを予約レジスタとして定義し、

上記使用レジスタ設定部は、使用可能レジスタを使用して実行するスレッドを作成するコンパイラを備えたことを特徴とする請求項1記載のオペレーティングシステムの処理方式。

【請求項3】 上記オペレーティングシステムの処理方式は、プロセッサ上に汎用レジスタを有し、

上記レジスタ設定定義ファイルは、定義された使用可能なレジスタを使用可能レジスタとして定義し、上記使用可能レジスタ以外のレジスタを予約レジスタとして定義し、

上記使用レジスタ設定部は、使用可能レジスタを使用するようにプロセッサの設定を行なうスレッド初期化処理部を備えたことを特徴とする請求項1記載のオペレーティングシステムの処理方式。

【請求項4】 上記オペレーティングシステムの処理方式は、さらに、上記予約レジスタの一部をオペレーティングシステムのグローバル変数に割り当てる設定を行なうオペレーティングシステム初期化処理部を備えたことを特徴とする請求項2または3記載のオペレーティングシステムの処理方式。

【請求項5】 上記オペレーティングシステムの処理方式は、さらに、上記予約レジスタの一部を特定のスレッドに割り当てる設定を行なうオペレーティングシステム初期化処理部を備え、

上記コンテキストスイッチ処理部は、上記特定のスレッドに割り当てられた予約レジスタの待避及び復帰は行なわず、上記特定のスレッドへ他のスレッドからコンテキストスイッチ処理を行なう場合には、他のスレッドが使用していた使用可能レジスタの待避を行ない、上記特定のスレッドから上記特定スレッド以外の他のスレッドへコンテキストスイッチ処理を行なう場合には、他のスレッドが使用していた使用可能レジスタの復帰を行うこと

を特徴とする請求項2または3記載のオペレーティングシステムの処理方式。

【請求項6】 上記オペレーティングシステムの処理方式は、さらに、スレッドが上記コンテキストスイッチ処理によって変更される回数をスレッド毎にカウントするスイッチカウンタを保持し、

上記スイッチカウンタを参照して特定スレッドを選択し、現在の特定スレッドと異なる場合には、特定スレッドを変更する優先スレッド選択処理部を備えたことを特徴とする請求項5記載のオペレーティングシステムの処理方式。

【請求項7】 上記オペレーティングシステムの処理方式は、さらに、上記予約レジスタの一部を割り込みにより起動される割り込みスレッドに割り当てる設定を行なう割り込み管理処理部を備え、

上記コンテキストスイッチ処理部は、上記割り込みスレッドへ他のスレッドからコンテキストスイッチ処理を行なう場合には、他のスレッドが使用していた使用可能レジスタの待避を行ない、上記割り込みスレッドから上記割り込みスレッド以外の他のスレッドへコンテキストスイッチ処理を行なう場合には、他のスレッドが使用していた使用可能レジスタの復帰を行うことを特徴とする請求項5記載のオペレーティングシステムの処理方式。

【請求項8】 上記レジスタ設定定義ファイルは、使用するレジスタを複数の分割して複数のレジスタセットを定義し、

上記使用レジスタ設定部は、複数の汎用レジスタを上記レジスタ設定ファイルで定義された複数のレジスタセットを分割し独立したスレッドに割り当てる設定をするオペレーション初期化処理部を備え、

上記コンテキストスイッチ処理部は、起動するスレッドが上記複数のレジスタセットのうちいずれかのレジスタセットを使用したスレッドであるかを判断し、上記複数のレジスタセットのうちいずれかのレジスタセットを使用したスレッド以外のスレッドを起動する場合は、レジスタに記憶されたデータの待避及び復帰をすることを特徴とする請求項1記載のオペレーティングシステムの処理方式。

【請求項9】 上記レジスタ設定定義ファイルは、使用するレジスタを二つに分割して二個のレジスタセットを定義することを特徴とする請求項8記載のオペレーティングシステムの処理方式。

【請求項10】 上記オペレーティングシステムの処理方式は、さらに、複数の浮動小数点レジスタと、浮動小数点レジスタを上記複数のレジスタセットに対応させて複数の分割して複数の浮動小数点レジスタセットを定義する浮動小数点レジスタ設定定義ファイルとを備え、

上記オペレーション初期化処理部は、さらに、上記レジスタ設定定義ファイルで定義されたレジスタセットに対

応して上記浮動小数点レジスタ設定定義ファイルで定義された浮動小数点レジスタセットを独立したスレッドに割り当てる設定を行ない、

上記コンテキストスイッチ処理部は、上記レジスタセットの待避及び復帰に伴って、対応する浮動小数点レジスタセットに記憶されたデータの待避及び復帰を行なうことを特徴とする請求項8記載のオペレーティングシステムの処理方式。

【請求項11】 処理を実行するスレッドが使用可能な複数の汎用レジスタを有するオペレーティングシステムの処理方法において、

汎用レジスタのうち使用するレジスタをレジスタ設定定義ファイルへ定義する工程と、

上記レジスタ設定定義ファイルを読み込み、スレッドが上記レジスタ設定定義ファイルで定義されたレジスタを使用するように設定する使用レジスタ設定工程と、

上記レジスタ設定定義ファイルで定義された使用するレジスタに記憶されたデータの待避、復帰を行なうコンテキストスイッチ処理を行なうコンテキストスイッチ処理工程とを備えたことを特徴とするオペレーティングシステムの処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、多数の汎用レジスタまたは浮動小数点レジスタを持つプロセッサを使用した情報処理装置上のオペレーティングシステム特にリアルタイムオペレーティングシステムにおけるスレッド間のコンテキストスイッチ時間の短縮化に関するオペレーティングシステムの処理方式及び処理方法である。

【0002】

【従来の技術】 従来、多数の汎用レジスタを持つプロセッサを使用した情報処理装置上のリアルタイムオペレーティングシステムの処理方式としては、例えば、特開平9-6630号公報に示されるようなものがあった。図21は、このようなリアルタイムオペレーティングシステムの汎用レジスタ管理の構成の一例を表す図である。

【0003】 多数の汎用レジスタは図21の1902a、1902bのようなレジスタウィンドウによって管理されており、実行中のスレッドが関数を呼び出す度に、レジスタウィンドウを切り替える。本従来例では、このレジスタウィンドウをユーザ用（「ユーザモード」ともいう）レジスタウィンドウ1902bとオペレーティングシステム用（「カーネルモード」ともいう）レジスタウィンドウ1902aに分けている。実行中のスレッドがユーザモードの場合は1902bのレジスタウィンドウ中の汎用レジスタを使い、カーネルモードに遷移した場合は1902aのレジスタウィンドウ中の汎用レジスタに切り替え、レジスタウィンドウ中の汎用レジスタを使う。このようにして、レジスタウィンドウを有効に利用して汎用レジスタの待避、復帰処理をなくし、ス

レッドのユーザモードからカーネルモードへの遷移を高速化することができる。

【0004】

【発明が解決しようとする課題】 上記のようなリアルタイムオペレーティングシステムの処理方式では、1スレッドのユーザモード、カーネルモード間の遷移は高速化できる。しかし、実行中のスレッドが他のスレッドへ実行を切り替えるコンテキストスイッチ処理には効果がないという問題点があった。従って、この方法ではスレッド間のコンテキストスイッチ時にはプロセッサの持つ全ての汎用レジスタの待避、復帰処理を行わなければならない、レジスタウィンドウのような多数の汎用レジスタを持つプロセッサの場合は、そのために大きなオーバーヘッドが発生するという問題点があった。

【0005】 また、上記のようなリアルタイムオペレーティングシステムの処理方式では、浮動小数点レジスタに対する考慮がない。このため、浮動小数点レジスタを多数持つプロセッサの場合、コンテキストスイッチ処理は、全ての浮動小数点レジスタの待避、復帰処理のために、さらに大きなオーバーヘッドが発生するという問題点があった。

【0006】 この発明は上記のような問題点を解消するためになされたもので、実行モジュールの使用する汎用レジスタを限定することにより、スレッド間のコンテキストスイッチにおいて、待避、復帰するレジスタ数を少なくし、コンテキストスイッチのオーバーヘッド時間を短縮することを目的とする。

【0007】 また、この発明は、プロセッサの持つレジスタウィンドウ制御機構、もしくは類似した機構を操作し、一部の汎用レジスタを使用しないようにすることにより、スレッド間のコンテキストスイッチにおいて、待避、復帰するレジスタ数を少なくし、コンテキストスイッチのオーバーヘッド時間を短縮することを目的とする。

【0008】 さらに、この発明は、上記の発明により、使用しないように設定した汎用レジスタを、頻繁にアクセスするオペレーティングシステムのグローバル変数として利用することにより、オペレーティングシステムの処理を高速化することを目的とする。

【0009】 この発明は、上記の発明により、使用しないように設定した汎用レジスタを、特定のスレッドのコンテキストとして固定的に割り当てることにより、特定スレッドへのコンテキストスイッチ時には、スレッドのコンテキストを持つ汎用レジスタの待避、復帰処理を行わずに、特定のスレッドへのコンテキストスイッチのオーバーヘッド時間を短縮することを目的とする。

【0010】 この発明は、上記の発明で、固定的に割り当てていた特定スレッドを、オペレーティングシステム実行中の各スレッドのスイッチ回数のカウントにより、もっともスイッチ回数の多いスレッドを動的に選択し、それを特定スレッドとすることにより、高速にコンテキ

ストスイッチが行える特定スレッドをオペレーティングシステムの動作状況に応じて有効に選択できるようにすることを目的とする。

【0011】この発明は、上記の発明により、使用しないように設定した汎用レジスタを、割込み処理より起動される割込みスレッドのコンテキストとして割り当て、割込みスレッド間で使い廻すようにすることにより、割込み発生時のコンテキストスイッチにおける汎用レジスタの待避、復帰処理をなくす、もしくは待避、復帰するコンテキスト量を限定し、割込み処理におけるコンテキストスイッチのオーバーヘッド時間を短縮することを目的とする。

【0012】この発明は、上記の発明により、汎用レジスタを2つのセットに分割し、それぞれに独立したスレッドのコンテキストを割り当て、コンテキストスイッチ処理において、前に実行したスレッドのコンテキストを保持したままにしておくことにより、2つのスレッド間の交互のコンテキストスイッチ時の汎用レジスタの待避、復帰処理をなくし、コンテキストスイッチのオーバーヘッド時間を短縮することを目的とする。

【0013】この発明は、上記発明の方法により、汎用レジスタをn個のセットに分割し、それぞれに独立したスレッドのコンテキストを割り当て、n個のレジスタセット間でそれぞれに独立したスレッドのコンテキストを割り当て、コンテキストスイッチ処理において、前に実行したスレッドのコンテキストを保持したままにしておく動作を行うことにより、n個のスレッド間のコンテキストスイッチ時の汎用レジスタの待避、復帰処理をなくし、コンテキストスイッチ時間を短縮することを目的とする。

【0014】この発明は、多数の浮動小数点レジスタを持つプロセッサ上のリアルタイムオペレーティングシステムに対し、それぞれに独立したスレッドのコンテキストを割り当て、コンテキストスイッチ処理において、前に実行したスレッドのコンテキストを保持したままにしておく動作を汎用レジスタだけでなく浮動小数点レジスタに対しても行い、浮動小数点コンテキストを持つスレッド間でのコンテキストスイッチ時の浮動小数点レジスタの待避、復帰処理を行う回数を減らし、コンテキストスイッチ時間を短縮することを目的とする。

【0015】

【課題を解決するための手段】この発明に係るオペレーティングシステムの処理方式は、処理を実行するスレッドが使用可能な複数の汎用レジスタを有するオペレーティングシステムの処理方式において、汎用レジスタのうち使用するレジスタを定義するレジスタ設定定義ファイルと、上記レジスタ設定定義ファイルを読み込み、スレッドが上記レジスタ設定定義ファイルで定義されたレジスタを使用するように設定する使用レジスタ設定部と、上記レジスタ設定定義ファイルで定義された使用するレ

ジスタに記憶されたデータの待避、復帰を行なうコンテキストスイッチ処理を行なうコンテキストスイッチ処理部とを備えたことを特徴とする。

【0016】上記レジスタ設定定義ファイルは、定義された使用可能なレジスタを使用可能レジスタとして定義し、上記使用可能レジスタ以外のレジスタを予約レジスタとして定義し、上記使用レジスタ設定部は、使用可能レジスタを使用して実行するスレッドを作成するコンパイラを備えたことを特徴とする。

【0017】上記オペレーティングシステムの処理方式は、プロセッサ上に汎用レジスタを有し、上記レジスタ設定定義ファイルは、定義された使用可能なレジスタを使用可能レジスタとして定義し、上記使用可能レジスタ以外のレジスタを予約レジスタとして定義し、上記使用レジスタ設定部は、使用可能レジスタを使用するようにプロセッサの設定を行なうスレッド初期化処理部を備えたことを特徴とする。

【0018】上記オペレーティングシステムの処理方式は、さらに、上記予約レジスタの一部をオペレーティングシステムのグローバル変数に割り当てる設定を行なうオペレーティングシステム初期化処理部を備えたことを特徴とする。

【0019】上記オペレーティングシステムの処理方式は、さらに、上記予約レジスタの一部を特定のスレッドに割り当てる設定を行なうオペレーティングシステム初期化処理部を備え、上記コンテキストスイッチ処理部は、上記特定のスレッドに割り当てられた予約レジスタの待避及び復帰は行なわず、上記特定のスレッドへ他のスレッドからコンテキストスイッチ処理を行なう場合には、他のスレッドが使用していた使用可能レジスタの待避を行ない、上記特定のスレッドから上記特定スレッド以外の他のスレッドへコンテキストスイッチ処理を行なう場合には、他のスレッドが使用していた使用可能レジスタの復帰を行うことを特徴とする。

【0020】上記オペレーティングシステムの処理方式は、さらに、スレッドが上記コンテキストスイッチ処理によって変更される回数をスレッド毎にカウントするスイッチカウンタを保持し、上記スイッチカウンタを参照して特定スレッドを選択し、現在の特定スレッドと異なる場合には、特定スレッドを変更する優先スレッド選択処理部を備えたことを特徴とする。

【0021】上記オペレーティングシステムの処理方式は、さらに、上記予約レジスタの一部を割込みにより起動される割込みスレッドに割り当てる設定を行なう割込み管理処理部を備え、上記コンテキストスイッチ処理部は、上記割込みスレッドへ他のスレッドからコンテキストスイッチ処理を行なう場合には、他のスレッドが使用していた使用可能レジスタの待避を行ない、上記割込みスレッドから上記割込みスレッド以外の他のスレッドへコンテキストスイッチ処理を行なう場合には、他のスレ

ッドが使用していた使用可能レジスタの復帰を行うことを特徴とする。

【0022】上記レジスタ設定定義ファイルは、使用するレジスタを複数に分割して複数のレジスタセットを定義し、上記使用レジスタ設定部は、複数の汎用レジスタを上記レジスタ設定ファイルで定義された複数のレジスタセットを分割し独立したスレッドに割り当てる設定をするオペレーション初期化処理部を備え、上記コンテキストスイッチ処理部は、起動するスレッドが上記複数のレジスタセットのうちいずれかのレジスタセットを使用したスレッドであるかを判断し、上記複数のレジスタセットのうちいずれかのレジスタセットを使用したスレッド以外のスレッドを起動する場合は、レジスタに記憶されたデータの待避及び復帰をすることを特徴とする。

【0023】上記レジスタ設定定義ファイルは、使用するレジスタを二つに分割して二個のレジスタセットを定義することを特徴とする。

【0024】上記オペレーティングシステムの処理方式は、さらに、複数の浮動小数点レジスタと、浮動小数点レジスタを上記複数のレジスタセットに対応させて複数に分割して複数の浮動小数点レジスタセットを定義する浮動小数点レジスタ設定定義ファイルとを備え、上記オペレーション初期化処理部は、さらに、上記レジスタ設定定義ファイルで定義されたレジスタセットに対応して上記浮動小数点レジスタ設定定義ファイルで定義された浮動小数点レジスタセットを独立したスレッドに割り当てる設定を行ない、上記コンテキストスイッチ処理部は、上記レジスタセットの待避及び復帰に伴って、対応する浮動小数点レジスタセットに記憶されたデータの待避及び復帰を行なうことを特徴とする。

【0025】この発明に係るオペレーティングシステムの処理方法は、処理を実行するスレッドが使用可能な複数の汎用レジスタを有するオペレーティングシステムの処理方法において、汎用レジスタのうち使用するレジスタをレジスタ設定定義ファイルへ定義する工程と、上記レジスタ設定定義ファイルを読み込み、スレッドが上記レジスタ設定定義ファイルで定義されたレジスタを使用するように設定する使用レジスタ設定工程と、上記レジスタ設定定義ファイルで定義された使用するレジスタに記憶されたデータの待避、復帰を行なうコンテキストスイッチ処理を行なうコンテキストスイッチ処理工程とを備えたことを特徴とする。

【0026】

【発明の実施の形態】実施の形態1. 以下、この発明の一例である実施の形態1を図に従って説明する。図1はこの発明の実施の形態1によるリアルタイムオペレーティングシステムの処理方式の一例を表した構成図である。101はレジスタ設定定義ファイルである。レジスタ設定定義ファイルは、「使用可能レジスタ」、「予約レジスタ」を定義する。102は本リアルタイムオペレ

ーティングシステム上で動作する実行モジュールすなわちスレッドを生成するコンパイラである。103はプロセッサの持つ汎用レジスタを示したものである。汎用レジスタはレジスタ設定定義ファイルにより、使用可能レジスタ103a、予約レジスタ103bに分けられる。104はオペレーティングシステムの持つコンテキストスイッチ処理部である。105は、コンパイラ102で生成した実行モジュールである。106はレジスタの割付に応じて汎用レジスタの名前を変更するプロセッサの持つレジスタリネーミング機構である。107は、上記レジスタ設定定義ファイルを読み込み、スレッドが上記レジスタ設定定義ファイルで定義されたレジスタを使用するように設定する使用レジスタ設定部である。また、この実施の形態では、使用レジスタ設定部として102に示すコンパイラである場合を説明する。尚、この明細書内では、レジスタに記憶されたデータを待避、復帰することを、レジスタを待避、復帰する、ということもあるが同様の動作を意味する。

【0027】この発明の実施の形態1の動作を示す。レジスタ設定定義ファイル101では、本リアルタイムオペレーティングシステム内でアプリケーションソフトウェアのスレッドが通常使用できるプロセッサの汎用レジスタを「使用可能レジスタ」として定義し、アプリケーションソフトウェアのスレッドが使用できないプロセッサの汎用レジスタを「予約レジスタ」として定義する。アプリケーションソフトウェアを生成するコンパイラ102はレジスタ設定定義ファイル101を参照し、「使用可能レジスタ」を使用するが「予約レジスタ」を使用しないアプリケーションソフトウェアの実行モジュールを作る。このように設定された状態で、オペレーティングシステムの持つコンテキストスイッチ処理部104の動作を図2に示す。コンテキストスイッチ処理部104は、コンテキストスイッチの初期化S101においてレジスタ設定定義ファイル101を参照し、「使用可能レジスタ」と「予約レジスタ」の定義を読み込む(S101)。これにより、以降でコンテキストスイッチ処理部104が呼び出されると、通常のコンテキストスイッチの処理の他にS102以下の動作を行う。ここで、通常のコンテキストスイッチ処理とは、現在実行中のスレッドから別のスレッドに切り替えることをいう。

【0028】S102では従来とは異なり、実行中のスレッドのコンテキストとなる汎用レジスタのうちの、「使用可能レジスタ」に設定されたレジスタに記憶されたデータをスタックに待避する。「予約レジスタ」に設定されたレジスタは待避しない。S103では従来通り、実行中のスレッドの持つ汎用レジスタ以外のコンテキストをスタックに待避する。S104ではS103に対応し、新しく起動するスレッドの汎用レジスタ以外のコンテキストをスタックから復帰する。S105ではS102に対応し、新しく起動するスレッドのコンテキ

トである汎用レジスタのうちの、「使用可能レジスタ」に設定されたレジスタをスタックから復帰する。この動作をコンテキストスイッチが発生するごとに繰り返す。

【0029】この実施の形態1によるオペレーティングシステム、特にリアルタイムオペレーティングシステムの処理方式によれば、アプリケーションソフトウェアのスレッドが使用する汎用レジスタ数を実行モジュールの生成時に限定するようにしたので、多数の汎用レジスタを持つプロセッサ上のリアルタイムオペレーティングシステムにおけるコンテキストスイッチ処理において、待避、復帰が必要なレジスタ数を少なくすることができ、コンテキストスイッチ処理のオーバーヘッド時間を削減することができる。

【0030】この発明のポイントは、コンテキストスイッチ処理部で実施するコンテキストスイッチ処理でレジスタを待避、または復帰するレジスタを限定することにある。従って、この実施の形態では、レジスタ定義設定ファイルにおいて、汎用レジスタを「使用可能レジスタ」と「予約レジスタ」の二種類に分割して定義する場合を示したが、「使用可能レジスタファイル」を汎用レジスタの一部に限定する定義であれば、「使用可能レジスタ」と「予約レジスタ」の二種類に分割する場合には限られない。従って、その他のレジスタを定義して二種類以上の定義を行なっても、「使用可能レジスタファイル」を汎用レジスタの一部に限定する定義であれば、本発明の実施の形態の一例として考えられる。また、上記のことは、以下の実施の形態2から6においても同様である。

【0031】以上のように、多数の汎用レジスタを持つプロセッサ上のリアルタイムオペレーティングシステムにおいて、使用する汎用レジスタ設定を定義するレジスタ設定定義ファイルとレジスタ設定定義ファイルを読み込み、レジスタ設定定義ファイルで「予約レジスタ」と指定された汎用レジスタを使用しない実行モジュールを生成するコンパイラと、汎用レジスタ設定定義ファイルを読み込み、汎用レジスタ設定定義ファイルで「予約レジスタ」と指定された汎用レジスタの待避、復帰を行わないコンテキストスイッチ処理部とを備えることによりリアルタイムオペレーティングシステムが実現できる。

【0032】実施の形態2。以下は、この発明の一例である実施の形態2を図に従って説明する。図3はこの発明の実施の形態2によるリアルタイムオペレーティングシステムの処理方式の構成の一例を表した図である。実施の形態2では、使用レジスタ設定部107として上記実施の形態1のコンパイラ102の代わりに、オペレーティングシステム内のスレッド初期化処理部201を追加する。スレッド初期化処理部は、「使用可能レジスタ」を使用するようにプロセッサの設定を行う。図3において、図1と同じ番号を持つものは、同じ構成要素である。

【0033】この発明の実施の形態2の動作を示す。実施の形態2では、アプリケーションソフトウェアを生成するコンパイラに代わり、オペレーティングシステム内のスレッド初期化処理部201がオペレーティングシステム起動時に、通常のスレッド初期化処理に加えて図4の動作を行う。ここで、通常のスレッド初期化処理とは、新しいスレッドを生成する処理をいう。具体的な処理はオペレーティングシステムによって異なる。S201でスレッド初期化処理部201はレジスタ設定定義ファイル101を読み込む。S202で、プロセッサの持つレジスタウィンドウを制御し、「予約レジスタ」に設定されたプロセッサの汎用レジスタを、レジスタウィンドウの割り当てから外す。S202の動作はレジスタウィンドウを持つプロセッサでなくても、プロセッサの持つ類似した汎用レジスタ割当て機構の制御により行ってもよい。これにより、アプリケーションソフトウェアのスレッドからは、「予約レジスタ」に設定されたレジスタを使用することがなくなる。この状態で、オペレーティングシステムのコンテキストスイッチ処理部104は、実施の形態1と同様に、図2で示した動作を行う。これによりコンテキストスイッチ処理において「予約レジスタ」に設定されたレジスタの待避、復帰を行わない。

【0034】この実施の形態2によるリアルタイムオペレーティングシステムの処理方式によれば、スレッド初期化時に、プロセッサのレジスタウィンドウに対する制御、または類似した汎用レジスタ割当て機構に対する制御により、アプリケーションソフトウェアのスレッドが使用する汎用レジスタを限定するようにしたので、多数の汎用レジスタを持つプロセッサ上のリアルタイムオペレーティングシステムにおけるコンテキストスイッチ処理において、待避、復帰が必要なレジスタ数を少なくすることができ、コンテキストスイッチ処理のオーバーヘッド時間を削減することができる。

【0035】以上のように、多数の汎用レジスタを持ち、関数呼出しごとに汎用レジスタの名前を変更し、アクセス可能な汎用レジスタを限定する機能を持つプロセッサ上のリアルタイムオペレーティングシステムにおいて、スレッドの初期化処理によりプロセッサを操作し、一部の汎用レジスタを使用しないようにプロセッサの設定を行うスレッド初期化処理部と、使用しないように設定した汎用レジスタの待避、復帰を行わないコンテキストスイッチ処理部とを備えることによりリアルタイムオペレーティングシステムを実施する。

【0036】実施の形態3。以下は、この発明の実施の形態3を図に従って説明する。図5はこの発明の実施の形態3によるリアルタイムオペレーティングシステムの処理方式の構成の一例を表す図である。実施の形態3では、上記実施の形態1、もしくは実施の形態2の構成に加え、オペレーティングシステム（以下、「OS」（O

perating System) ともいう) の初期化処理部301を追加する。また、オペレーティングシステム処理内で頻繁にアクセスするOSグローバル変数302を定義する。図5に表していない構成要素については実施の形態1、もしくは実施の形態2の構成と同様の構成となるが、図5では省略している。

【0037】実施の形態3では、実施の形態1、または実施の形態2の動作と同様に、プロセッサの持つ汎用レジスタの一部を「予約レジスタ」と設定し、アプリケーションソフトウェアのスレッドに使用させない。また、コンテキストスイッチ処理においても、「予約レジスタ」と設定したレジスタの待避、復帰処理を行わない。実施の形態3では、この「予約レジスタ」に対し、オペレーティングシステムの初期化処理部(図5では「OS初期化処理部」と表示)301において、オペレーティングシステム全体の通常の初期化処理の他に図6の動作を行う。オペレーティングシステム処理内で頻繁にアクセスし性能上影響を与えるグローバル変数のいくつかをOSグローバル変数302として定義する。S301では、このOSグローバル変数302を「予約レジスタ」にコピーする。これ以降、OSグローバル変数302に対するアクセスは、「予約レジスタ」に対し行う。

【0038】この実施の形態3によるリアルタイムオペレーティングシステムの処理方式によれば、実施の形態1、または実施の形態2の特徴に加え、OSグローバル変数302を常にレジスタ参照によりアクセスできるので、プロセッサの持つ多数の汎用レジスタを有効に利用でき、オペレーティングシステムの動作を高速化することができる。

【0039】以上のように、リアルタイムオペレーティングシステムにおいて、「予約レジスタ」としてリザーブした汎用レジスタを、頻繁にアクセスするオペレーティングシステムのグローバル変数に割り当てるように設定するオペレーティングシステムの初期化処理を持ち、設定したグローバル変数に対しては、オペレーティングシステム内の全て処理において、メモリでなく割り当てたレジスタを直接参照することによりリアルタイムオペレーティングシステムを実現する。

【0040】実施の形態4。以下、この発明の実施の形態4を図に従って説明する。図7はこの発明の実施の形態4によるリアルタイムオペレーティングシステムの処理方式の構成の一例を表す図である。実施の形態4では、実施の形態1、または実施の形態2に加え、オペレーティングシステムの初期化処理部(図7では「OS初期化処理部」と表示)401、コンテキストスイッチ処理部402を変更する。また、オペレーティングシステム設計時にリアルタイムオペレーティングシステムの実行性能上重要なスレッドを一つ選択する。OS初期化処理部401は、選択されたスレッドを特定スレッドとしてスレッドのコンテキスト403を別管理する。また、

106はプロセッサの汎用レジスタの名前付けを変更するレジスタリネーミング機構である。図7に表していない構成要素については実施の形態1、もしくは実施の形態2の構成と同様の構成となるが、図7では省略している。

【0041】実施の形態4では、実施の形態1、または実施の形態2の動作と同様に、プロセッサの持つ汎用レジスタの一部を「予約レジスタ」と設定し、通常のアプリケーションソフトウェアのスレッドに使用させない。実施の形態4では、OS初期化処理部401において、選択された特定スレッドのコンテキスト403を「予約レジスタ」に固定的に割り当てる(図8のS401)。特定スレッドは、オペレーティングシステムの設計時に、性能上重要となるスレッドを1つ固定的に選択する。この後、コンテキストスイッチ処理部402は図9の動作を行う。コンテキストスイッチ処理は、初期化処理において従来のコンテキストスイッチ処理の他に、S402でレジスタ設定定義ファイルを読み込み、実施の形態1または実施の形態2と同様に、プロセッサの汎用レジスタの「使用可能レジスタ」と「予約レジスタ」の設定を行う。それ以降、コンテキストスイッチ処理が発生すると、S403で現在実行中のスレッドが特定スレッドか否かをチェックする。実行中のスレッドが特定スレッドでない場合には、従来どおりS404で実行中のスレッドのコンテキストとなる汎用レジスタのうちの、「使用可能レジスタ」に設定されたレジスタをスタックに待避する。特定スレッドの場合には、S404の動作を行わず、代わりにS410で、プロセッサの使用するレジスタを「予約レジスタ」から「使用可能レジスタ」に変更する。S410の処理は後述するS409の逆処理で、プロセッサの持つレジスタリネーミング機構106、または類似した機構により、特定スレッド実行中に使用しているレジスタである「予約レジスタ」の名前を元の「予約レジスタ」に戻し、「使用可能レジスタ」の名前を「使用可能レジスタ」に戻すことにより実現する。次に従来通りS405で、実行中のスレッドの汎用レジスタ以外のその他のコンテキストをスタックに待避する。S406ではS405と対応して、新しく起動するスレッドの汎用レジスタ以外のコンテキストをスタックから復帰する。S407では新しく起動するスレッドが特定スレッドかどうかをチェックし、新しく起動するスレッドが特定スレッドでない場合には、S408で起動するスレッドのコンテキストである汎用レジスタのうちの、「使用可能レジスタ」に設定されたレジスタをスタックから復帰する。S407で新しく起動するスレッドが特定スレッドの場合には、S408の動作を行わず、S409で「予約レジスタ」の方を使用レジスタに設定し、コンテキストスイッチ処理を完了させる。S409の処理はプロセッサの持つレジスタリネーミング機構106、または類似した機構により、「予約レジ

タ」の汎用レジスタ名を変更し、「予約レジスタ」の名前を「使用可能レジスタ」と同じにするとともに、「使用可能レジスタ」の名前を「予約レジスタ」のレジスタの名前にすることにより実現する。これにより、特定スレッドも通常のスレッドと同じ汎用レジスタの名前を使うことができる。

【0042】この実施の形態4によるリアルタイムオペレーティングシステムの処理方式によれば、実施の形態1、または実施の形態2の特徴に加え、特定スレッドのコンテキストを常に汎用レジスタ上に保持するようにしたので、特定スレッドへのコンテキストスイッチ、または特定スレッドからその他のスレッドへのコンテキストスイッチ時には、汎用レジスタ内のコンテキストの待避、復帰処理が不要になり、コンテキストスイッチ時間が短縮できる。

【0043】以上のように、リアルタイムオペレーティングシステムにおいて、「予約レジスタ」としてリザーブした汎用レジスタを、特定のスレッドのコンテキストに固定的に割り当てるOS初期化処理を持つ。選択した特定のスレッドへコンテキストスイッチする場合には、汎用レジスタの「使用可能レジスタ」と指定したレジスタの待避のみを行い、復帰処理を行わないコンテキストスイッチ処理を持つ。一方、選択した特定のスレッドから別のスレッドへコンテキストスイッチする場合には、汎用レジスタの「使用可能レジスタ」と指定したレジスタの待避処理を行わず、復帰処理のみを行うコンテキストスイッチ処理を持つ。このようにしてリアルタイムオペレーティングシステムを実現する。

【0044】実施の形態5。以下、この発明の実施の形態5を図に従って説明する。図10はこの発明の実施の形態5によるリアルタイムオペレーティングシステムの処理方式の構成の一例を表す図である。実施の形態5では、実施の形態4の構成に加え、一定時間ごとにサービスを行う周期起床処理を呼び出すタイマ処理501、特定スレッドを決定する手段である優先スレッド選択処理部502、各スレッドがコンテキストスイッチ処理によって変更されるスイッチ回数を記録するスイッチカウンタ503を持つ。スイッチカウンタ503は、スレッドID503aとカウンタ503bで構成された表である。504はオペレーティングシステム内に存在するスレッドのコンテキストのスレッド汎用レジスタを示している。505は実施の形態5におけるコンテキストスイッチ処理部である。図10に表していない構成要素については実施の形態1、もしくは実施の形態2の構成と同様の構成となるが、図10では省略している。

【0045】実施の形態5のコンテキストスイッチ処理部505では、実施の形態4のコンテキストスイッチ処理部402の図9の動作に対し、図11のように処理を加える。図11中、図9と同じ符号の処理は、同一のものを表している。具体的には、S402からS410の

動作の後で、S511でスイッチカウンタ503内の、新しく起動するスレッドを示すスレッドID503aに対応するカウンタ503bをカウントアップする。これにより、各スレッドのスイッチ回数を保持する。また、特定スレッドの選択方法は、実施の形態4のOS初期化処理で行うのではなく、図12に示す優先スレッド選択処理部502で実施する優先スレッド選択処理の動作により行う。優先スレッド選択処理部502は、タイマ処理501より、一定周期でごとに呼び出される。優先スレッド選択処理部502はS501で、スレッドカウンタを参照し、もっともスイッチ回数が多いスレッドを選択する。さらに、S502で、選択したスレッドが現在の特定スレッドと同じかどうかをチェックする。選択したスレッドが現在の特定スレッドと異なる場合には、S503で、「予約レジスタ」に入っている特定スレッドのコンテキストを待避し、代わりに新しく選択したスレッドのコンテキストを「予約レジスタ」に復帰する。S502で、選択したスレッドが現在の特定スレッドと同じ場合には、何もしない。この動作をタイマ処理501より呼び出されるごとに繰り返す。この動作と実施の形態4の動作により、特定スレッドとその他のスレッド間のコンテキストスイッチにおける汎用レジスタの待避、復帰処理をなくす。

【0046】この実施の形態5によるリアルタイムオペレーティングシステムの処理方式では、実施の形態4において固定的に行っていた特定スレッドの選択を、オペレーティングシステム実行中の各スレッドのスイッチカウンタでカウントしているスイッチ回数のカウントにより、もっともスイッチ回数の多いスレッドを動的に選択するようにしたので、高速にコンテキストスイッチを行える特定スレッドをオペレーティングシステムの動作状況に応じて有効に選択することができる。

【0047】以上のように、リアルタイムオペレーティングシステムにおいて、オペレーティングシステム内で動作しているスレッドがコンテキストスイッチ処理により変換されるスイッチ回数を保持する各スレッドごとのスイッチカウンタを持つ。スイッチカウンタを参照し、もっとも値の大きいものを特定スレッドとして選択し、上記実施の形態4の動作を行わせる優先スレッド選択処理部を持つ。また優先スレッド選択処理部の動作をオペレーティングシステム動作中に定期に実行するタイマ処理を持つ。このようにしてリアルタイムオペレーティングシステムを実現する。

【0048】実施の形態6。以下、この発明の実施の形態6を図に従って説明する。図13はこの発明の実施の形態6によるリアルタイムオペレーティングシステムの処理方式の構成図である。実施の形態6では、実施の形態4に加え、割り込み処理より起動されるスレッドである割り込みスレッドのコンテキスト管理を行う割り込み管理処理部601を持つ。603は割り込みにより直接起動され

る割込みハンドラ、604は本実施の形態におけるコンテキストスイッチ処理部である。602は割込み処理を行うスレッド、すなわち割込みスレッドのコンテキストを示す。また、割込みスレッドは割込みハンドラ603からの継続処理として呼び出されるスレッドで、オペレーティングシステム内で別管理し、割込みスレッド用コンテキスト602内にコンテキストを保持しておく。図13に表していない構成要素については実施の形態1、もしくは実施の形態2の構成と同様の構成となるが、図13では省略している。

【0049】実施の形態6では、割込みハンドラ603の継続処理として呼び出される割込みスレッド用コンテキスト602を割込み管理処理部601で別に管理する。実施の形態6におけるコンテキストスイッチ処理部604、および割込み管理処理部601の動作を図14に示す。コンテキストスイッチ処理部604は、実施の形態4と同様に初期化処理において、S601でレジスタ設定定義ファイルを読み込み、プロセッサの汎用レジスタの「使用可能レジスタ」と「予約レジスタ」の設定を行う。それ以降、コンテキストスイッチ処理部604は、コンテキストスイッチ処理が発生すると、S602で割込み発生による割込みスレッドへのコンテキストスイッチかをチェックする。S602で割込みスレッドへのコンテキストスイッチの場合には、割込み管理処理601を呼び出し、それにより処理を行う。S602でそれ以外のスレッドへのコンテキストスイッチの場合には、S609で、現在「使用可能レジスタ」を保持しているのが、これから新しく起動するスレッドかどうかをチェックする。「使用可能レジスタ」を保持しているのが新しく起動するスレッドの場合は、コンテキストの復帰の必要がないのでコンテキストスイッチ処理を完了する。S609で、「使用可能レジスタ」を保持しているのが新しく起動するスレッドでない場合には、従来と同様に、S610で現在実行中のスレッドの汎用レジスタの内容をスタックへ待避し、S611で汎用レジスタ以外のコンテキストを待避し、S612で新しく起動するスレッドのS611に対応したその他のコンテキストをスタックから復帰し、S613で新しく起動するスレッドのS613に対応した汎用レジスタの内容をスタックから復帰し、コンテキストスイッチ処理を完了する。

【0050】S602で、新しく起動するスレッドが割込みスレッドの場合は、割込み管理処理部601を呼び出し、以下のように動作する。まず、S603で現在「予約レジスタ」を保持しているのは、新しく起動する割込みスレッドかをチェックする。「予約レジスタ」を保持しているのが新しく起動する割込みスレッドでない場合は、S604で現在の「予約レジスタ」の内容をスタックへ待避する。さらにS605で起動する割込みスレッドの汎用レジスタのコンテキストを「予約レジスタ」に復帰した後、S606の動作を行う。S603で

「予約レジスタ」を保持しているのが新しく起動する割込みスレッドの場合は、S604、S605の動作は行わず、S606に移る。S606では「予約レジスタ」の方を使用するレジスタに設定する。この処理は、レジスタリネーミング機構106、または類似した機構により、「予約レジスタ」の汎用レジスタ名を変更し、「予約レジスタ」の名前を「使用可能レジスタ」と同じにすることにより実現する。その後、S607では割込みスレッドの処理を実行する。割込みスレッドの処理実行完了後、S608で「使用可能レジスタ」の方を使用するレジスタに設定する。この処理は、S606の逆操作として、レジスタリネーミング機構106、または類似した機構により「予約レジスタ」、「使用可能レジスタ」の名前を元に戻すことにより実現する。これにより、割込み管理処理でのコンテキストスイッチ処理が完了する。このコンテキストスイッチ処理部604、割込み管理処理601の動作を、コンテキストスイッチが必要になるごとに繰り返す。

【0051】この実施の形態6によるリアルタイムオペレーティングシステムの処理方式によれば、実施の形態4では特定の1つのスレッドへのコンテキストスイッチが高速に行えるのに対し、リアルタイムオペレーティングシステムの実行性能上重要となる割込みスレッドのコンテキストを「予約レジスタ」に保持するようにしたので、割込みスレッドへのスイッチが高速に行えるようになる。さらに、割込みスレッドのコンテキストは「予約レジスタ」のみを使用するので、待避、復帰が必要な汎用レジスタ数が少なく、複数の割込みスレッド間でコンテキストスイッチが必要になり、「予約レジスタ」の内容を切り替える場合でも、通常のスレッド間のコンテキストスイッチよりも少ないオーバーヘッドで行うことができる。割込みスレッドの処理は、通常のアプリケーションソフトウェアのスレッドよりも処理が単純で短いことが想定されるので、大量のレジスタを割り当てなくても性能低下がない。

【0052】以上のように、リアルタイムオペレーティングシステムにおいて、「予約レジスタ」によってリザーブした汎用レジスタを、割込み処理より起動されるシステムスレッドに割り当てる割込み管理処理部を持つ。割込み処理用の割込みスレッドへコンテキストスイッチする場合は、「予約レジスタ」と指定した汎用レジスタのみの待避、復帰を行い、他の汎用レジスタの待避、復帰を行わないコンテキストスイッチ処理を持つ。一方、割込み処理用の割込みスレッドから、割込まれたスレッドへ復帰する場合は、汎用レジスタの待避、復帰を行わないコンテキストスイッチ処理を持つ。このようにしてリアルタイムオペレーティングシステムを実現する。

【0053】実施の形態7. 以下、この発明の実施の形態7を図に従って説明する。図15はこの発明の実施の形態7によるリアルタイムオペレーティングシステムの

処理方式の構成の一例を表す図である。実施の形態7では、使用レジスタ設定部107として、オペレーティングシステムの初期化処理部(図15では「OS初期化処理」と表示)705を備える。また、プロセッサの汎用レジスタ702を2つに分割し、レジスタセット1(702a)と、レジスタセット2(702b)とする。またレジスタセットの定義のため、レジスタ設定定義ファイル701では、それぞれのレジスタセットを構成するレジスタ名を定義する。コンテキストスイッチ処理部703はこの2つのレジスタセットを管理する。また、704aは現在実行中のスレッドのコンテキスト、704bは次に新しく起動するスレッドのコンテキストである。106はレジスタリネーミング機構であり、実施の形態1または2と同様のものである。

【0054】実施の形態7では、実施の形態1、または実施の形態2の方法と同様に、レジスタ設定定義ファイル701で定義し、オペレーティングシステムの初期化処理部705で、プロセッサの持つ汎用レジスタ702を2つに分割し、レジスタセット1(702a)とレジスタセット2(702b)とする。さらに、オペレーティングシステムの初期化処理部705は、この2つのレジスタセットをそれぞれ独立したスレッドに割り当てられるように設定する。レジスタセット1(702a)、レジスタセット2(702b)は、レジスタリネーミング機構106により汎用レジスタの名前を変更することにより、どちらか一方を使用するレジスタに設定する。この2つのレジスタセットを使い、コンテキストスイッチ処理部703は図16のように動作する。コンテキストスイッチ処理部703は、初期化処理においてS701でレジスタ設定定義ファイルを読み込み、レジスタセット1、レジスタセット2の構成定義を設定する。それ以降、コンテキストスイッチ処理が発生すると、以下のような動作を行う。ここではレジスタセット1を現在使用している汎用レジスタとして説明する。S702で使用していないレジスタセットであるレジスタセット2が保持している内容が、これから新たに起動するスレッドのコンテキストであるかをチェックする。レジスタセット2が保持している内容が新たに起動するスレッドのコンテキストでない場合には、S705でレジスタセット2の内容をスタックに待避する。次にS706で現在実行中のスレッドの汎用レジスタ以外のコンテキストをスタックへ待避する。S707でS706と対応する新たに起動するスレッドの汎用レジスタ以外のコンテキストをスタックから復帰する。S708で起動するスレッドの汎用レジスタのコンテキストを現在使用していないレジスタセットであるレジスタセット2に復帰する。その後S709で、使用するレジスタセットをレジスタセット1からレジスタセット2に切り替えることにより、コンテキストスイッチ処理を完了する。使用するレジスタセットの切り替えはレジスタリネーミング機構106、

または類似した機構により、レジスタセット2の汎用レジスタ名をレジスタセット1のものに変更することにより実現する。

【0055】この処理ではレジスタセット1にはまだ実行中だったスレッドのコンテキストを保持している。S702で、レジスタセット2の内容が新たに起動するスレッドのコンテキストである場合には、S703で、S706と同様に、現在実行中のスレッドの汎用レジスタ以外のコンテキストをスタックへ待避する。次にS704で、S707と同様に、新たに起動するスレッドの汎用レジスタ以外のコンテキストをスタックから復帰する。この後、S709に戻り、使用するレジスタセットをレジスタセット1からレジスタセット2に切り替える。この場合には汎用レジスタの待避、復帰処理は行っていない。使用している汎用レジスタがレジスタセット2の場合には、上記の説明のレジスタセット1とレジスタセット2を読み替える。これらの動作により、レジスタセット1とレジスタセット2にはコンテキストスイッチによりスイッチする前のスレッドのコンテキスト704aと、スイッチした後のスレッドのコンテキスト704bが保持されていることになる。スイッチした後のスレッドからスイッチする前のスレッドに戻る場合は、汎用レジスタの待避、復帰処理が不要になる。

【0056】この実施の形態7によるリアルタイムオペレーティングシステムの処理方式によれば、コンテキストスイッチ前後のスレッドのコンテキストを2つのレジスタセットで持つようにしたので、あるスレッドが別のスレッドに実行をプリエンプションされた後、元のスレッドに復帰するような2つのスレッド間で交互に発生するコンテキストスイッチなどの実行の遷移を、汎用レジスタの待避、復帰処理を行わずに実現でき、コンテキストスイッチ処理のオーバーヘッド時間を短縮することができる。

【0057】この発明のポイントは、コンテキストスイッチ処理部で実施するコンテキストスイッチ処理でレジスタを待避、または復帰するレジスタを限定することにある。この実施の形態では、一つのスレッドが使用できる「使用可能レジスタ」のレジスタの数を限定し、更に、一つのスレッドが使用できる「使用可能レジスタ」のセットを二つ定義した点に特徴がある。従って、この実施の形態では、レジスタ定義設定ファイルにおいて、汎用レジスタを二つのレジスタセットに分割して定義する場合を示したが、二つのレジスタセット以外のレジスタが存在しても構わない。

【0058】以上のようにリアルタイムオペレーティングシステムにおいて、汎用レジスタを「使用可能レジスタ」と「予約レジスタ」とに分けることにより、汎用レジスタを2つのセットに分割し、それぞれに独立したスレッドコンテキストを持たせるオペレーティングシステム初期化処理を持つ。コンテキストスイッチを行う場

合、現在使用していない汎用レジスタセットの保持しているコンテキストの内容を待避し、そのレジスタセットに対し起動するスレッドのコンテキストの復帰を行い、現在使用している汎用レジスタセットの内容はそのまま保持し、使用する汎用レジスタセットを変更するコンテキストスイッチ処理を持つ。一方、コンテキストスイッチを行う場合、現在使用していない汎用レジスタセットに保持しているコンテキストの所有者が、これから起動するスレッドである場合には、汎用レジスタの待避、復帰を行わないコンテキストスイッチ処理を持つ。このようにしてリアルタイムオペレーティングシステムを実現する。

【0059】実施の形態8. 以下、この発明の実施の形態8を図に従って説明する。図17はこの発明の実施の形態8によるリアルタイムオペレーティングシステムの処理方式の構成の一例を表す図である。実施の形態8では、実施の形態7における汎用レジスタの分割を2個からn個に変更する。これによりレジスタ設定定義ファイル801では、1からnまでのレジスタセットの構成定義を設定する。オペレーティングシステムの初期化処理部705は実施の形態7と同様にレジスタセットの設定を行う。プロセッサの汎用レジスタ802はレジスタ設定定義ファイル801により、レジスタセット1(802a)、レジスタセット2(802b)、レジスタセット3(802c)、・・・、レジスタセットn(802n)、というように分割される。複数のレジスタセットに分割する方法は、実施の形態1、または実施の形態2の方法を応用し、選択したレジスタセットを「使用可能レジスタ」、それ以外のレジスタセットを「予約レジスタ」とし、その「使用可能レジスタ」と「予約レジスタ」の構成をレジスタセットの選択を変えるごとに変更することにより実現する。コンテキストスイッチ処理部803はこれらのレジスタセットを管理する。804a、804b、804c、・・・、は各レジスタセットに保持されているスレッドのコンテキストである。

【0060】実施の形態8のコンテキストスイッチ処理部803は図18のように動作する。コンテキストスイッチ処理部803は、初期化処理においてS801でレジスタ設定定義ファイル801を読み込み、レジスタセット1からレジスタセットnまでの構成定義を設定する。それ以降、コンテキストスイッチ処理が発生すると、以下のような動作を行う。S802で、新たに起動するスレッドのコンテキストがレジスタセット1からレジスタセットnまでに割り当てられているかをチェックする。新たに起動するスレッドのコンテキストがレジスタセットのどれかに割り当てられている場合には、S803で現在実行中のスレッドの汎用レジスタ以外のコンテキストをスタックに待避し、S804で新たに起動するスレッドの汎用レジスタ以外のコンテキストをスタックから復帰する。この場合は、汎用レジスタの待避、復

帰処理は行わない。その後、S810を行う。S802で新たに起動するスレッドのコンテキストがレジスタセットに割り当てられていない場合は、S805でレジスタセット1からレジスタセットnの中から一つのレジスタセットを選択する。

【0061】本実施の形態では、もっとも最近使用していないレジスタセットを選択しているが、他のアルゴリズムによりレジスタセットを選択しても良い。その後は実施の形態7の動作と同様に、S806で選択したレジスタセットの内容をスタックに待避し、S807で現在実行中のスレッドの汎用レジスタ以外のコンテキストをスタックへ待避し、S808で新たに起動するスレッドの汎用レジスタ以外のコンテキストをスタックから復帰し、S809で新たに起動するスレッドの汎用レジスタのコンテキストを、S805で選択したレジスタセットに復帰する。S810で、使用レジスタセットを現在実行中のスレッドのコンテキストを保持しているレジスタセットから、S805で選択したレジスタセットに切り替える。使用するレジスタセットの変更の方法はレジスタリネーミング機構106、または類似した機構により、汎用レジスタ名を切り替えることにより実現する。これにより、スレッドのコンテキスト804a、804b、804c、・・・804nがキャッシングされることになり、このスレッド間でコンテキストスイッチする場合は、汎用レジスタの待避、復帰処理が不要になる。

【0062】この実施の形態8によるリアルタイムオペレーティングシステムの処理方式によれば、汎用レジスタをn個のレジスタセットに分割し、それをスレッドの汎用レジスタのコンテキストのキャッシュとするようにしたので、実施の形態7の特徴である2つのスレッド間での交互に発生するコンテキストスイッチの高速化を、n個のスレッド間で実現でき、その間の実行の遷移では、汎用レジスタの待避、復帰処理を行わないことにより、コンテキストスイッチ処理のオーバーヘッド時間を短縮することができる。

【0063】この発明のポイントは、コンテキストスイッチ処理部で実施するコンテキストスイッチ処理でレジスタを待避、または復帰するレジスタを限定することにある。この実施の形態では、一つのスレッドが使用できる「使用可能レジスタ」のレジスタの数を限定し、更に、一つのスレッドが使用できる「使用可能レジスタ」のセットを複数定義した点に特徴がある。従って、この実施の形態では、レジスタ定義設定ファイルにおいて、汎用レジスタを複数のレジスタセットに分割して定義する場合を示したが、定義したレジスタセット以外のレジスタが存在しても構わない。また、上記のことは、以下の実施の形態9においても同様である。

【0064】実施の形態9. 以下、この発明の実施の形態9を図に従って説明する。図19はこの発明の実施の形態9によるリアルタイムオペレーティングシステムの

処理方式の構成の一例を表す図である。実施の形態9では、実施の形態8の構成に加え、プロセッサが多数の浮動小数点レジスタ901を保持することを想定している。浮動小数点レジスタは、実施の形態1、または実施の形態2と同様の方法を用いて、浮動小数点レジスタ設定定義ファイル905に示される定義に従い、 n 個のレジスタセット901a、901b、901c、・・・、901nに分割する。オペレーティングシステムの初期化処理部906は、レジスタ設定定義ファイル801と浮動小数点レジスタ設定定義ファイル905に基づいて、汎用レジスタのレジスタセットを浮動小数点のレジスタセットの設定を行う。コンテキストスイッチ処理部902では、実施の形態8の汎用レジスタのレジスタセット802a、802b、802c、・・・、803nに加え、浮動小数点レジスタのレジスタセット901a、901b、901c、・・・、901nを管理する。904は106と同様に浮動小数点レジスタの名前を制御するために、プロセッサの持つ浮動小数点レジスタリネーミング機構である。オペレーティングシステム内のスレッドは、浮動小数点演算を行うものに関しては、汎用レジスタのコンテキストの他に、浮動小数点レジスタのコンテキストを保持する。図19の例では、スレッド1とスレッド3は浮動小数点レジスタコンテキストも持つコンテキスト903a、903cを保持し、スレッド2は汎用レジスタコンテキストのみを持つコンテキスト903bを保持している。903a、903b、903c、・・・、は汎用レジスタセット、浮動小数点レジスタセットに保持されているスレッドのコンテキストである。

【0065】実施の形態9のコンテキストスイッチ処理部902は図20のように動作する。コンテキストスイッチ処理部902は、初期化処理においてS901でレジスタ設定定義ファイル801と、浮動小数点レジスタ設定定義ファイル905を読み込み、汎用レジスタ802のレジスタセット1からレジスタセット n までの構成定義と、浮動小数点レジスタ901のレジスタセット1からレジスタセット n までの構成定義を設定する。それ以降、コンテキストスイッチ処理が発生すると、以下のような動作を行う。S902で新たに起動するスレッドが浮動小数点レジスタコンテキストを保持しているかどうかチェックする。起動するスレッドが浮動小数点レジスタコンテキストを保持していない場合は、図18のS802からS810の動作を行い、実施の形態8と同様にコンテキストスイッチ処理を行う。S902で新たに起動するスレッドが浮動小数点レジスタコンテキストを保持している場合には、S903で、新たに起動するスレッドの浮動小数点レジスタコンテキストが、浮動小数点レジスタのレジスタセット1からレジスタセット n までに割り当てられているかをチェックする。新たに起動するスレッドの浮動小数点レジスタコンテキストが浮動

小数点レジスタのレジスタセットのどれかに割り当てられている場合は、浮動小数点レジスタの待避、復帰処理は行わず、S907を行う。S903で新たに起動するスレッドの浮動小数点レジスタコンテキストが浮動小数点レジスタのレジスタセットに割り当てられていない場合は、S904で浮動小数点のレジスタセット1からレジスタセット n の中から一つのレジスタセットを選択する。

【0066】本実施の形態では、もっとも最近使用していない浮動小数点レジスタセットを選択しているが、他のアルゴリズムによりレジスタセットを選択しても良い。次にS905で選択した浮動小数点レジスタのレジスタセットをスタックへ待避し、S906で新たに起動するスレッドの浮動小数点レジスタコンテキストを、S904で選択したレジスタセットに復帰する。S907で使用する浮動小数点レジスタのレジスタセットを、現在、浮動小数点レジスタを保持しているスレッドのレジスタセットから、S905で選択したレジスタセットに切り替える。使用する浮動小数点レジスタのレジスタセットの変更は、汎用レジスタの場合と同様に、浮動小数点レジスタリネーミング機構904、または類似した機構により、浮動小数点レジスタ名を切り替えることにより実現する。この処理の後、S908に移り、実施の形態8の動作により、汎用レジスタのコンテキストの切り替え処理を行う。これにより、汎用レジスタに加え、浮動小数点レジスタコンテキストも903a、903c、・・・、903nのようにキャッシングされることになり、このスレッド間でコンテキストスイッチする場合は、浮動小数点レジスタの待避、復帰処理も不要になる。尚、この実施の形態では浮動少数点レジスタについて説明したが、この他のレジスタに適用できることは言うまでもない。

【0067】この実施の形態9によるリアルタイムオペレーティングシステムの処理方式では、実施の形態8に加え、浮動小数点レジスタも n 個のレジスタセットに分割し、それを浮動小数点レジスタコンテキストのキャッシュとするようにしたので、 n 個の浮動小数点レジスタコンテキストを持つスレッド間でコンテキストスイッチする場合には、浮動小数点コンテキストの待避、復帰処理もなくなり、実施の形態8よりもさらにコンテキストスイッチ処理のオーバーヘッド時間を短縮することができる。

【0068】以上のように、多数の浮動小数点レジスタを持つプロセッサ上のリアルタイムオペレーティングシステムにおいて、浮動小数点レジスタに対しても本発明の処理方式を適用し、浮動小数点レジスタの一部を使用しないようにし、浮動小数点レジスタを使用するスレッドに対し、上記実施の形態8の動作を汎用レジスタと共に浮動小数点レジスタに対して行うオペレーティングシステムの初期化処理とコンテキストスイッチ処理を持

つ。このようにしてリアルタイムオペレーティングシステムを実現する。

【0069】実施の形態10。上記実施の形態1～9では、リアルタイムオペレーションシステムについて説明したが、オペレーションシステムであれば、この処理方式を適用することも可能である。また、上記の実施の形態1～9では、汎用レジスタのうち、使用する汎用レジスタを制限することにより、コンテキストスイッチ処理のオーバーヘッド時間の短縮する処理方式について説明したが、汎用レジスタ以外のレジスタについてこの処理方式を適用することも可能である。更に、レジスタのみでなく、この他のメモリ等についても、この処理方式を適用することも考えられる。

【0070】

【発明の効果】この発明に係るオペレーティングシステムの処理方式及びオペレーティングシステムの処理方法によれば、実行モジュールの使用する汎用レジスタを限定することにより、スレッド間のコンテキストスイッチにおいて、待避、復帰するレジスタ数を少なくし、コンテキストスイッチのオーバーヘッド時間を短縮することができる。

【0071】さらに、この発明によれば、一部の汎用レジスタを使用しないようにすることにより、スレッド間のコンテキストスイッチにおいて、待避、復帰するレジスタ数を少なくし、コンテキストスイッチのオーバーヘッド時間を短縮することができる。

【0072】この発明によれば、使用しないように設定した汎用レジスタを、頻繁にアクセスするオペレーティングシステムのグローバル変数として利用することにより、オペレーティングシステムの処理を高速化することができる。

【0073】この発明によれば、特定スレッドへのコンテキストスイッチ時には、スレッドのコンテキストを持つ汎用レジスタの待避、復帰処理を行わずに、特定のスレッドへのコンテキストスイッチのオーバーヘッド時間を短縮することができる。

【0074】この発明によれば、高速にコンテキストスイッチが行える特定スレッドをオペレーティングシステムの動作状況に応じて有効に選択できるようにすることができる。

【0075】この発明によれば、割り込み発生時のコンテキストスイッチにおける汎用レジスタの待避、復帰処理をなくす、もしくは待避、復帰するコンテキスト量を限定し、割り込み処理におけるコンテキストスイッチのオーバーヘッド時間を短縮することができる。

【0076】この発明によれば、複数個のスレッド間のコンテキストスイッチ時の汎用レジスタの待避、復帰処理をなくし、コンテキストスイッチ時間を短縮することができる。

【0077】この発明によれば、2つのスレッド間の交

互のコンテキストスイッチ時の汎用レジスタの待避、復帰処理をなくし、コンテキストスイッチのオーバーヘッド時間を短縮することができる。

【0078】この発明によれば、浮動小数点コンテキストを持つスレッド間でのコンテキストスイッチ時の浮動小数点レジスタの待避、復帰処理を行う回数を減らし、コンテキストスイッチ時間を短縮することができる。

【図面の簡単な説明】

【図1】 この発明の実施の形態1によるリアルタイムオペレーティングシステムの処理方式の構成の一例を表した図である。

【図2】 この発明の実施の形態1のコンテキストスイッチ処理部で実施する処理のフロー図である。

【図3】 この発明の実施の形態2によるリアルタイムオペレーティングシステムの処理方式の構成の一例を表した図である。

【図4】 この発明の実施の形態2のスレッド初期化処理部で実施するスレッド初期化処理のフロー図である。

【図5】 この発明の実施の形態3によるリアルタイムオペレーティングシステムの処理方式の構成の一例を表す図である。

【図6】 この発明の実施の形態3のOS初期化処理部で実施するOS初期化処理のフロー図である。

【図7】 この発明の実施の形態4によるリアルタイムオペレーティングシステムの処理方式の構成の一例を表す図である。

【図8】 この発明の実施の形態4のOS初期化処理部で実施するOS初期化処理のフロー図である。

【図9】 この発明の実施の形態4のコンテキストスイッチ処理部で実施する処理のフロー図である。

【図10】 この発明の実施の形態5によるリアルタイムオペレーティングシステムの処理方式の構成の一例を表す図である。

【図11】 この発明の実施の形態5のコンテキストスイッチ処理部で実施する処理のフロー図である。

【図12】 この発明の実施の形態5の優先スレッド選択処理部で実施する優先スレッド選択処理のフロー図である。

【図13】 この発明の実施の形態6によるリアルタイムオペレーティングシステムの処理方式の構成図である。

【図14】 この発明の実施の形態6のコンテキストスイッチ処理部と割り込み管理処理部で実施する処理のフロー図である。

【図15】 この発明の実施の形態7によるリアルタイムオペレーティングシステムの処理方式の構成の一例を表す図である。

【図16】 この発明の実施の形態7のコンテキストスイッチ処理部で実施する処理のフロー図である。

【図17】 この発明の実施の形態8によるリアルタイ

ムオペレーティングシステムの処理方式の構成の一例を表す図である。

【図18】 この発明の実施の形態8のコンテキストスイッチ処理部で実施する処理のフロー図である。

【図19】 この発明の実施の形態9によるリアルタイムオペレーティングシステムの処理方式の構成の一例を表す図である。

【図20】 この発明の実施の形態9のコンテキストスイッチ処理部で実施する処理のフロー図である。

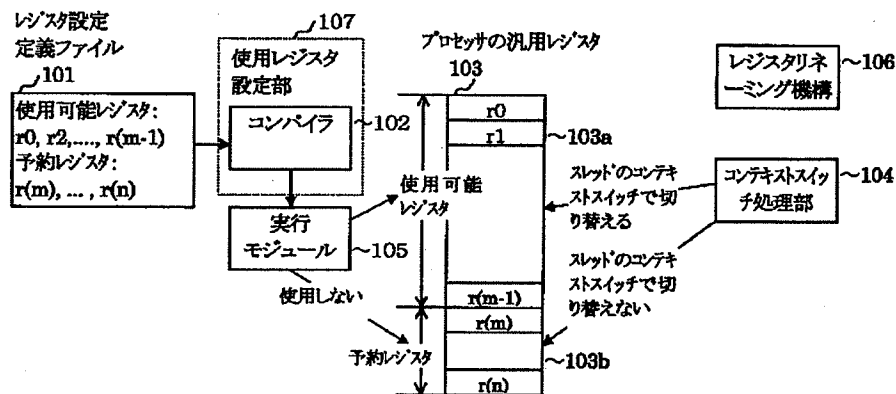
【図21】 従来のリアルタイムオペレーティングシステムの汎用レジスタ管理の構成の一例を表す図である。

【符号の説明】

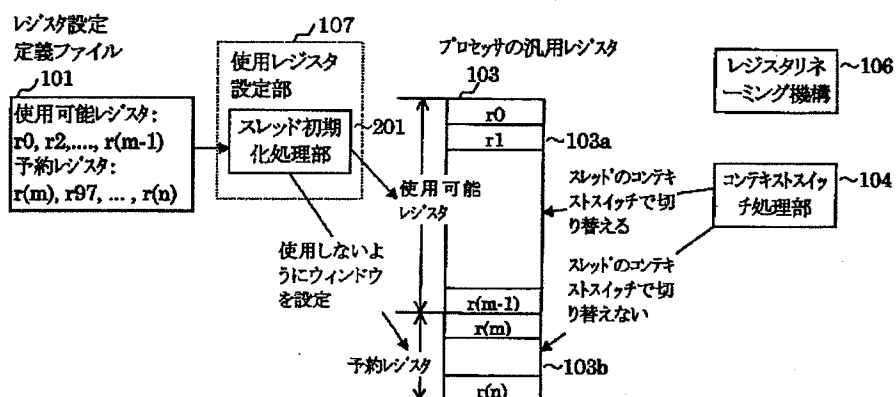
101, 701, 801 レジスタ設定定義ファイル、102 コンパイラ、103, 103a, 103b, 702, 702a, 702b, 802, 802a~802n、プロセッサの汎用レジスタ、104, 402, 505, 604, 703, 803, 902 コンテキストスイッチ処理部、105 実行モジュール、106 レジ

スタリネーミング機構、107 使用レジスタ設定部、201スレッド初期化処理部、301, 401, 705, 906 オペレーティングシステムの初期化処理部（OS初期化処理部）、302 OSグローバル変数、403 特定スレッド用汎用レジスタ、501 タイマ処理、502 優先スレッド選択処理部、503 スイッチカウンタ、503a スレッドID、503bカウンタ、504 スレッド汎用レジスタ、601 割込み管理処理部、602 割込みスレッド用コンテキスト、603 割込ハンドラ、704a 現実行スレッドコンテキスト、704b スイッチするスレッドコンテキスト、804a~804n スレッドコンテキスト、901, 901a~901n 浮動小数点レジスタ、904 浮動小数点レジスタリネーミング機構、905 浮動小数点レジスタ設定定義ファイル、1901, 1902a, 1902b レジスタウィンドウ、1903a, 1903b 無効マスク、1904 カレントウィンドウポインタ。

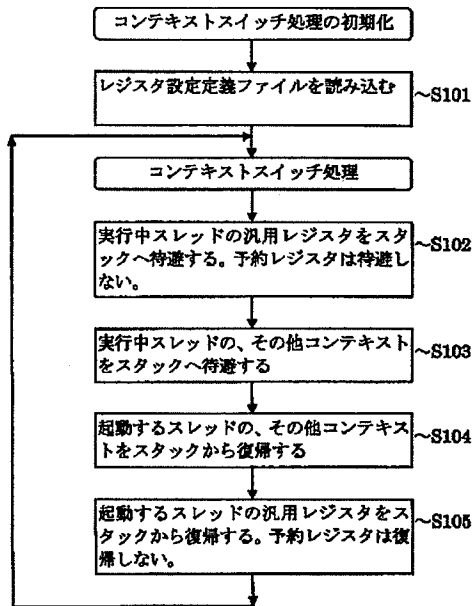
【図1】



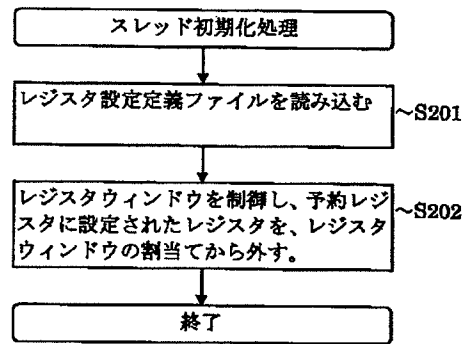
【図3】



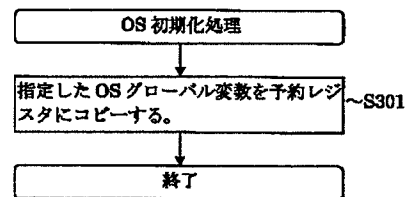
【図2】



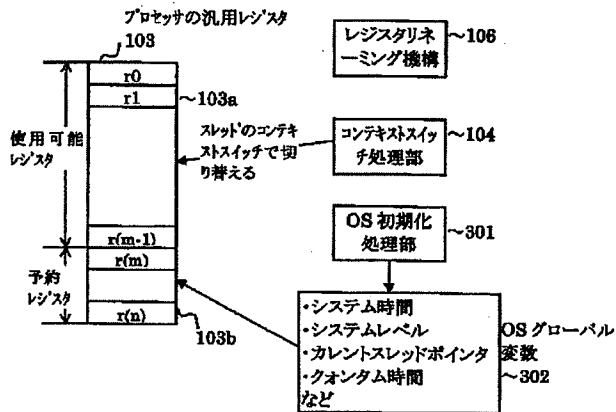
【図4】



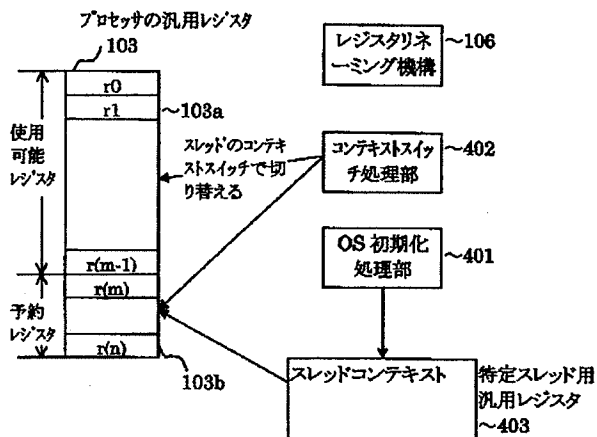
【図6】



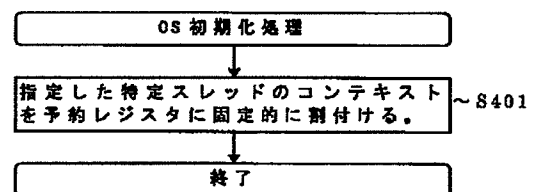
【図5】



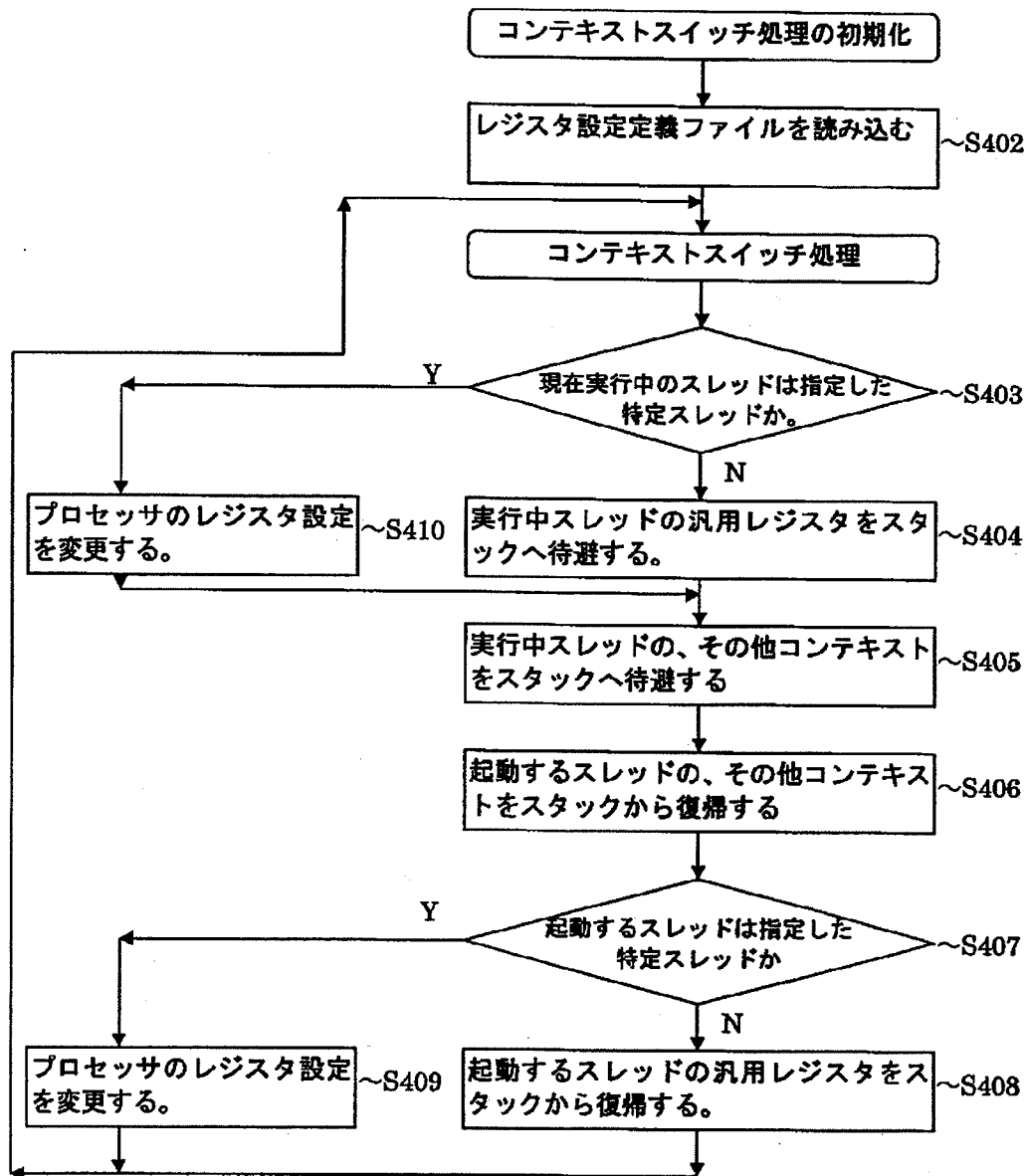
【図7】



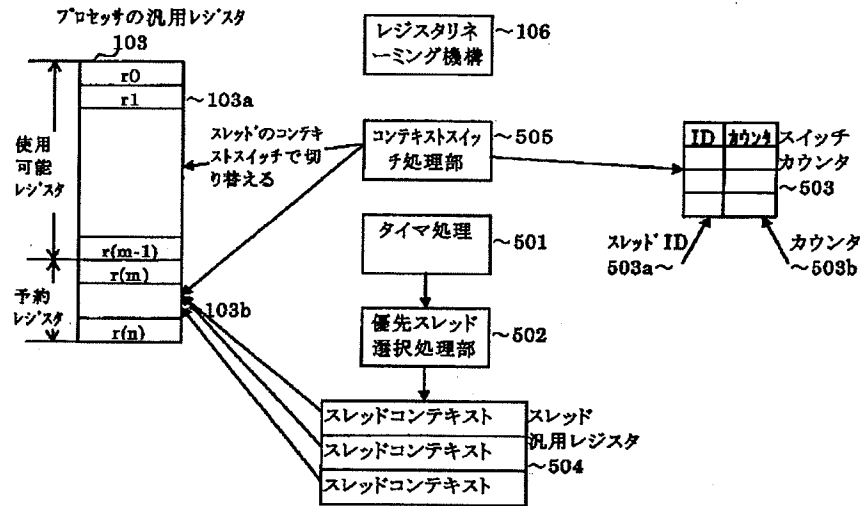
【図8】



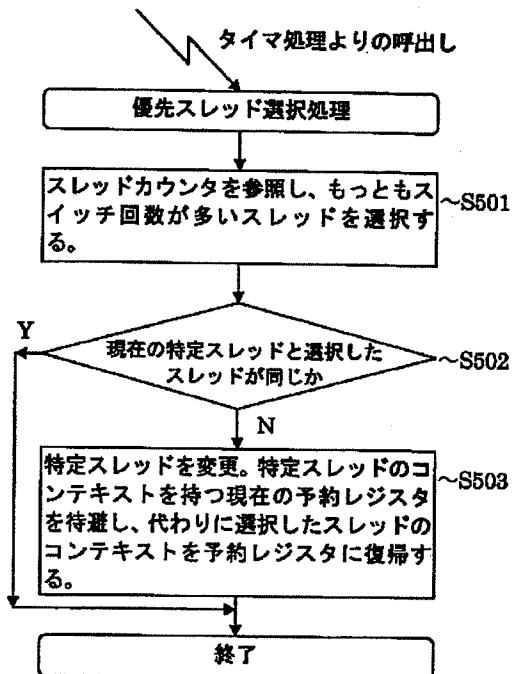
【図9】



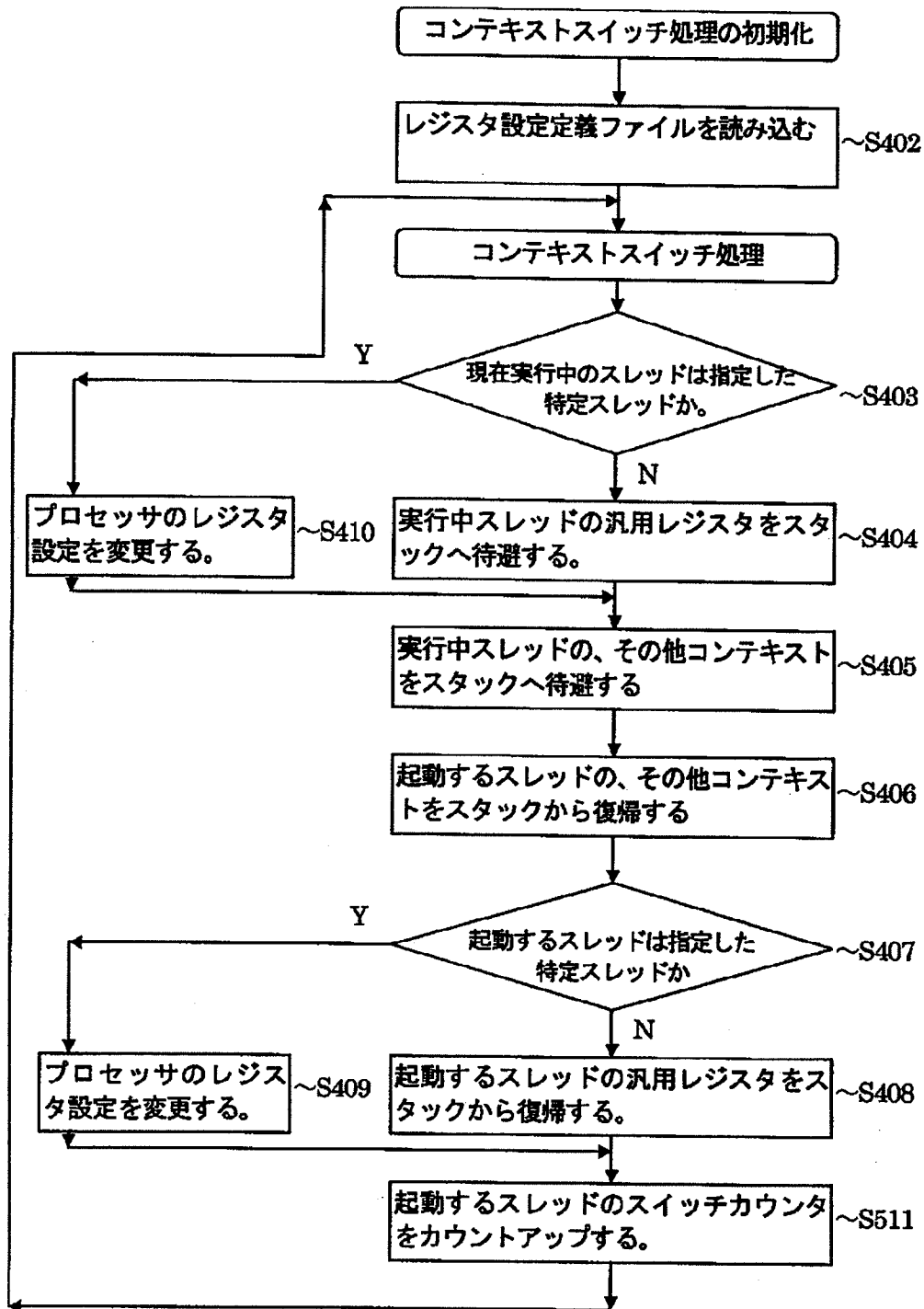
【図10】



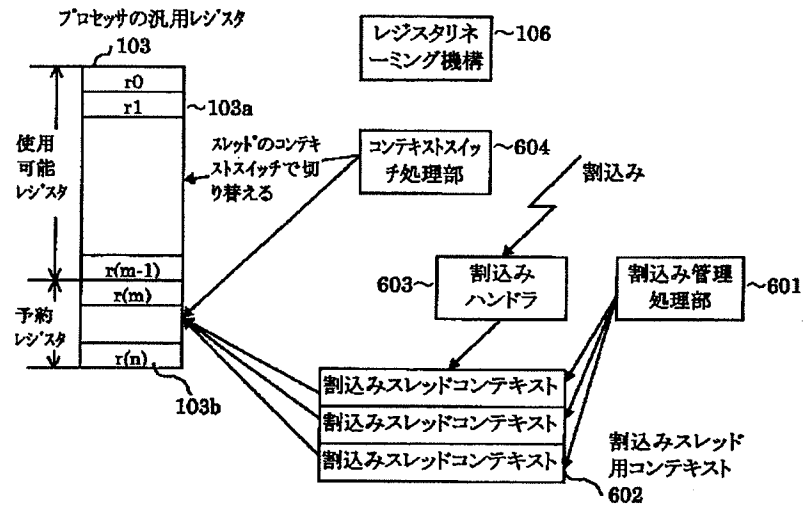
【図12】



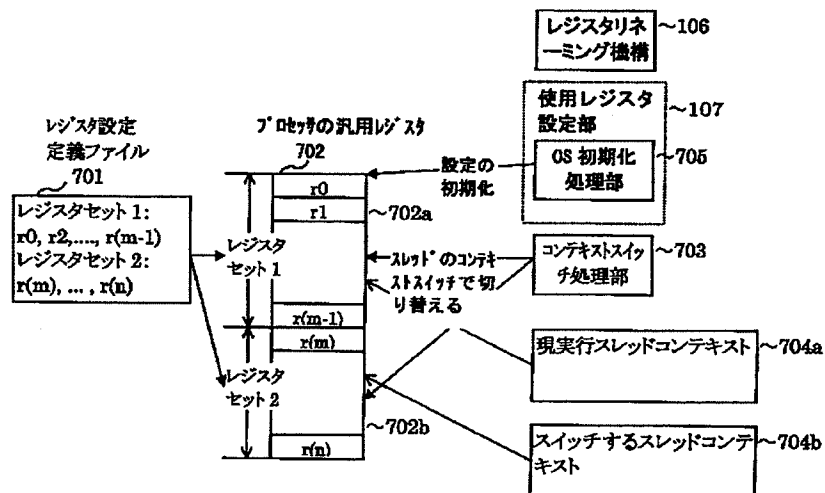
【図11】



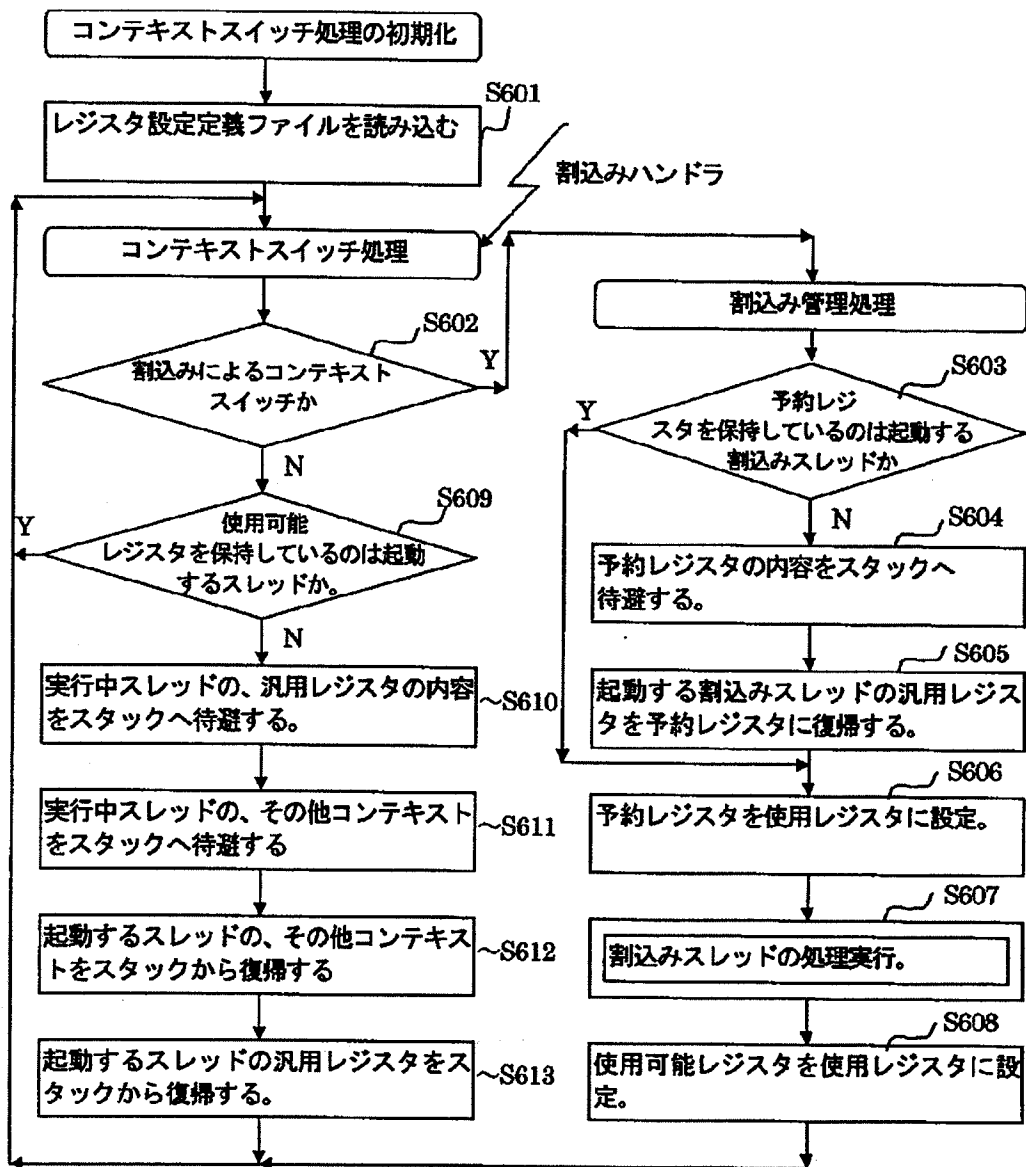
【図13】



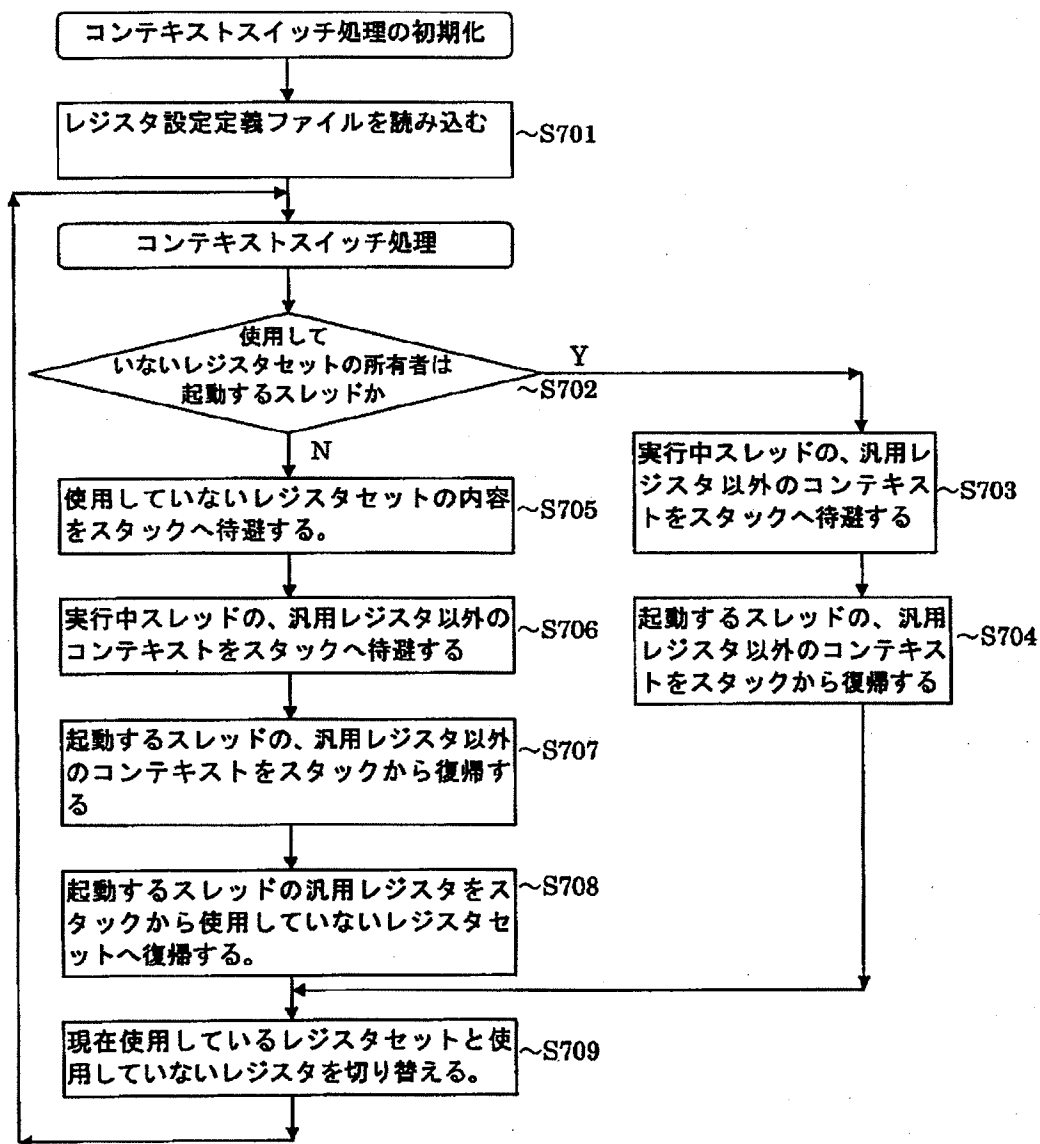
【図15】



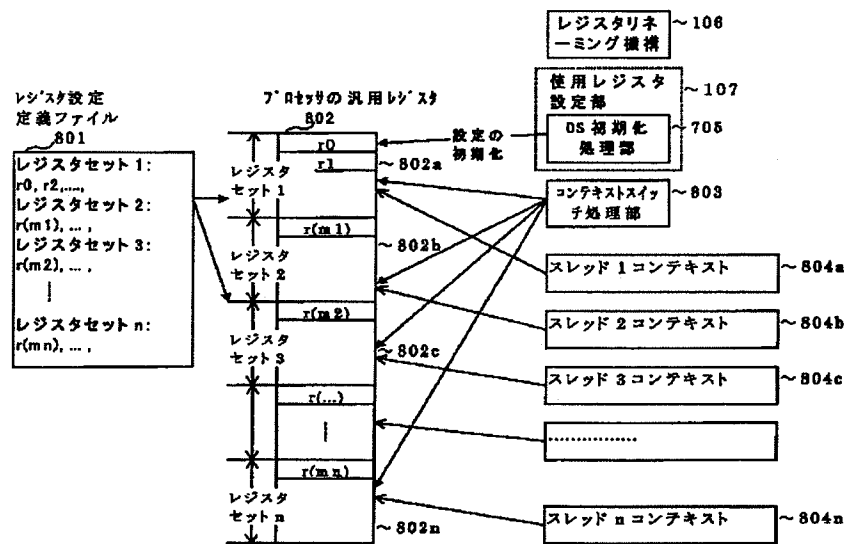
【図14】



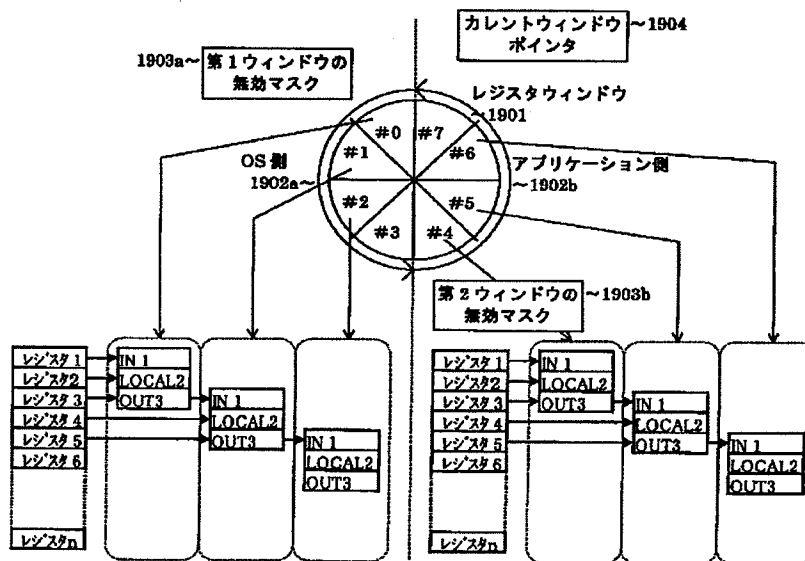
【図16】



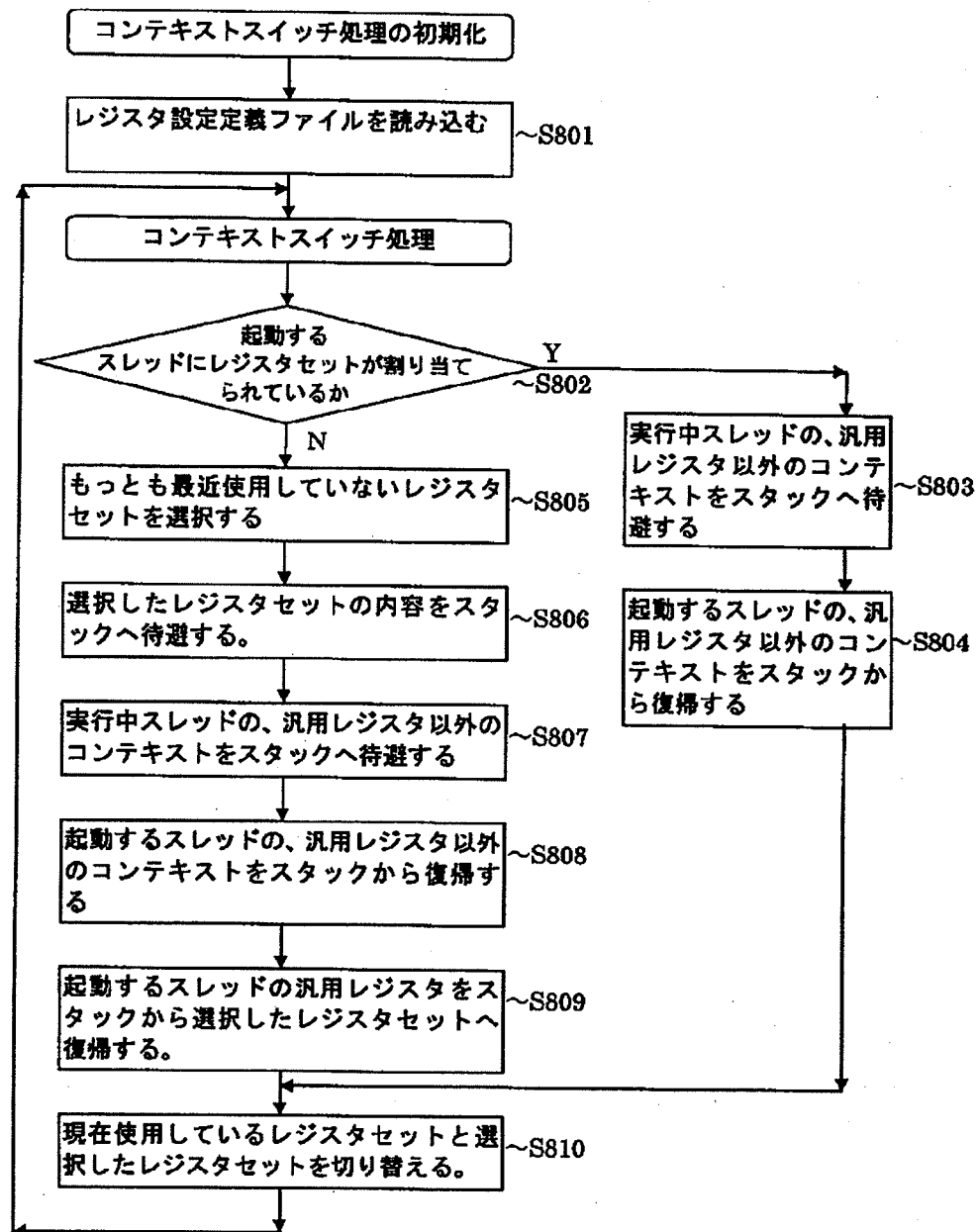
【図17】



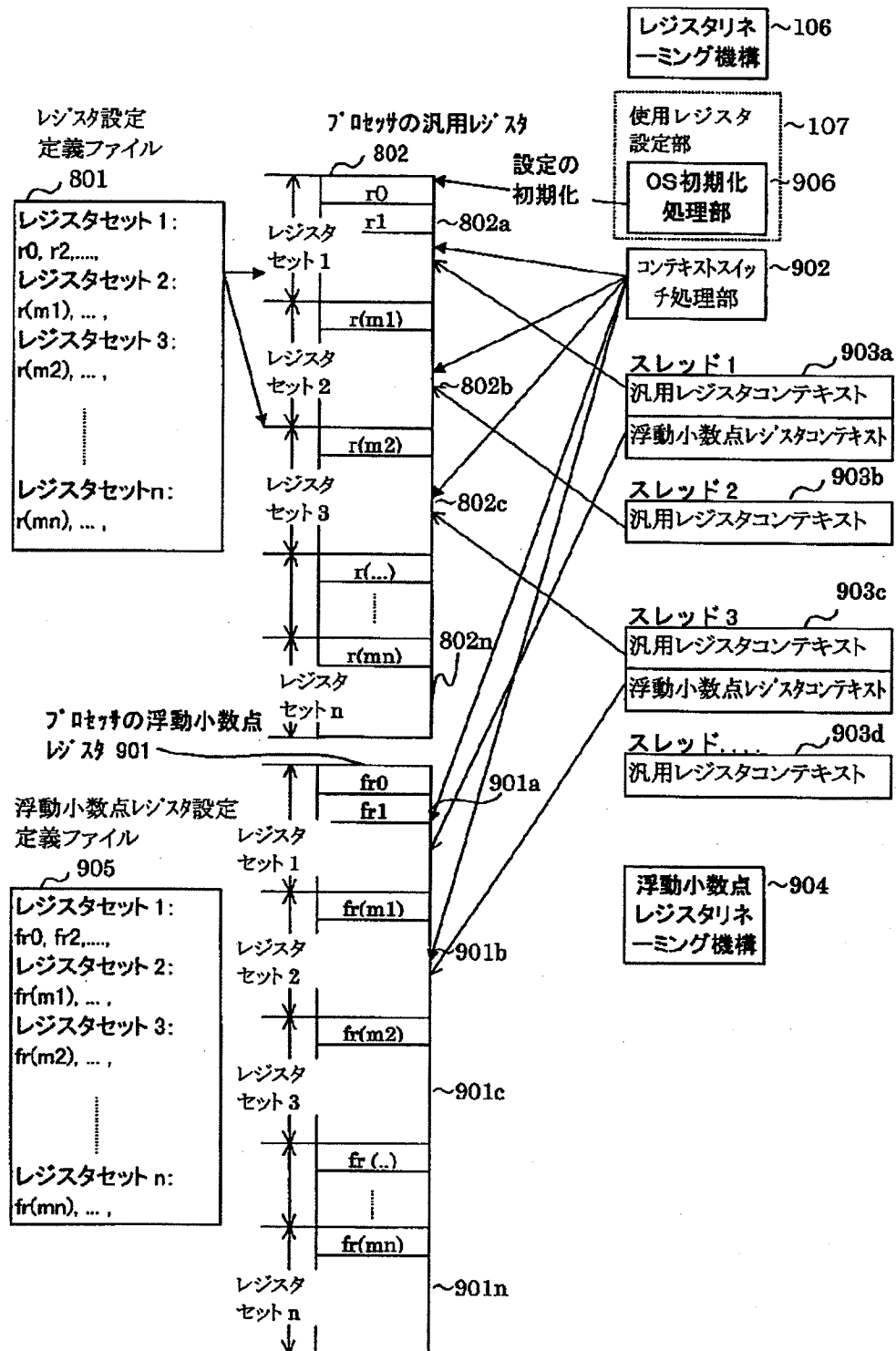
【図21】



【図18】



【図19】



【図20】

